

[54] CIRCLE OR ARC GENERATOR FOR GRAPHIC DISPLAY

[75] Inventors: Lorenza S. Childress, Salisbury Mills; Ralph J. Gundrum, Rhineback, both of N.Y.

[73] Assignee: International Business Machines Corporation, Armonk, N.Y.

[22] Filed: June 30, 1972

[21] Appl. No.: 267,800

[52] U.S. Cl.: 235/151, 235/198, 340/324 A

[51] Int. Cl.: G06f 3/14

[58] Field of Search: 235/151, 151.11, 197, 198; 340/324 A

[56] References Cited

UNITED STATES PATENTS

3,702,922	11/1972	Hall, Jr. et al.	235/151
3,696,391	10/1972	Peronneau	235/198
3,675,231	7/1972	Bezrodny	340/324 A
3,594,756	7/1971	Granberg et al.	340/324 A
3,555,538	1/1971	Henderson et al.	340/324 A
3,335,416	8/1967	Hughes	340/324 A
3,199,111	8/1965	Jennings et al.	235/151 X

Primary Examiner—Felix D. Gruber

Attorney, Agent, or Firm—W. S. Robertson

[57] ABSTRACT

The operations in a data processing system for drawing circles or other geometric figures on a display which are conventionally defined by a series of instructions prepared by a problem programmer are included in a standardized form in a small memory that is made part of the display unit. The problem programmer has only to specify either the starting point of an arc of the center point of a circle, the radius of the arc or circle, and other optional characteristics such as whether the circle or arc is to be drawn clockwise or counter clockwise, whether the beam of the display is blanked or unblanked, and whether additional instructions for drawing additional arc segments are to be expected or requested by the display. Logic and timing means are provided for accessing the memory in response to these values specified by the problem programmer and for generating a desired sequence of X and Y incremental signals for operating the display.

8 Claims, 4 Drawing Figures

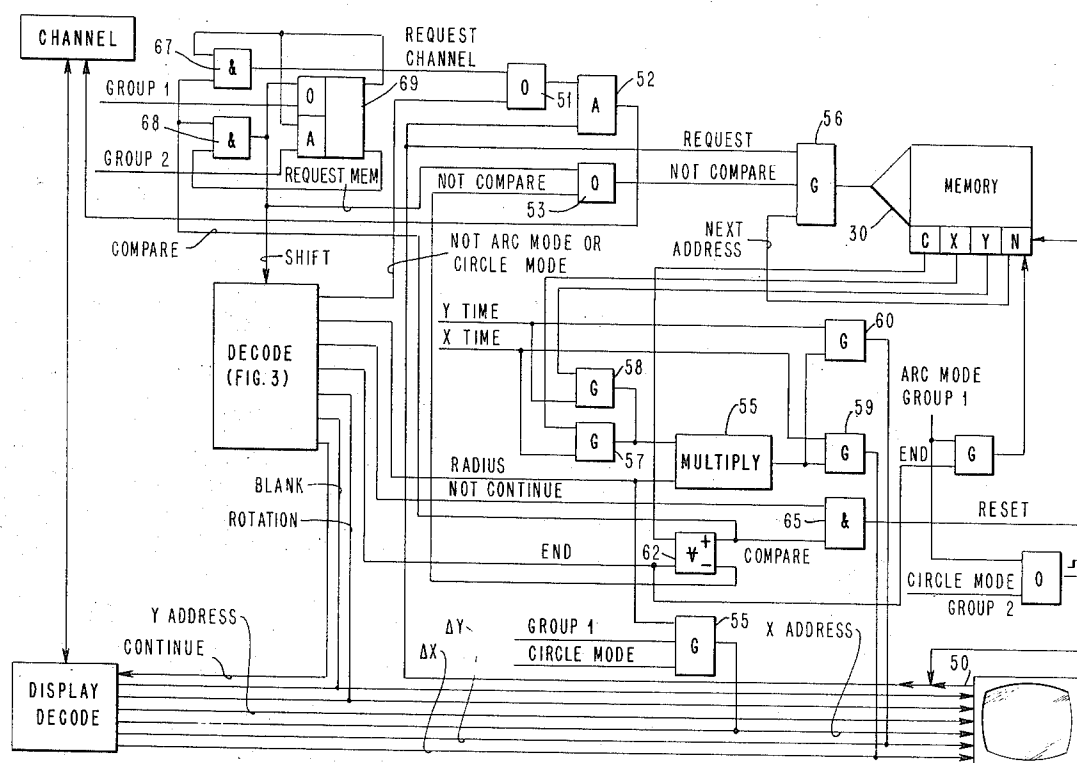
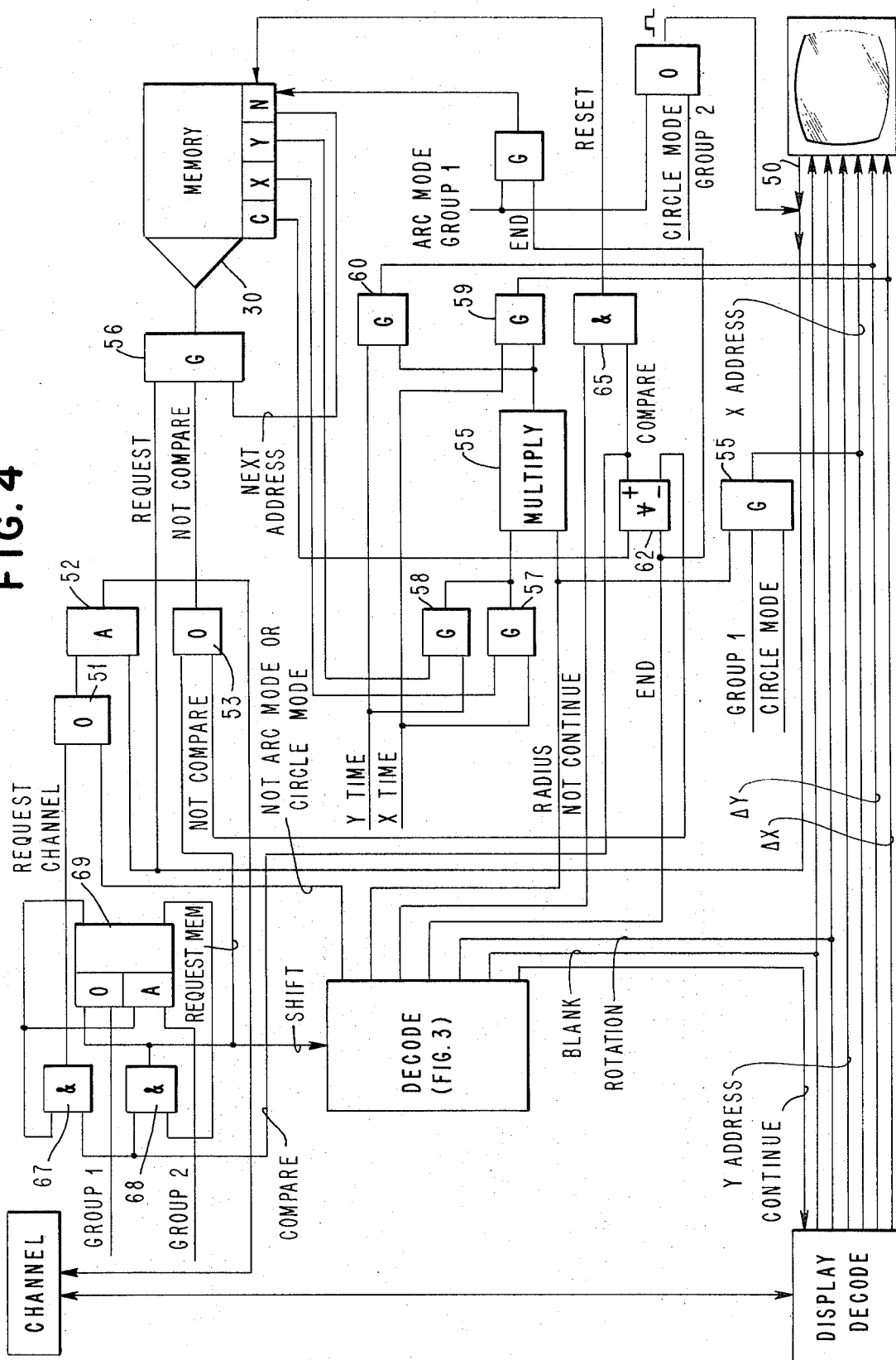


FIG. 4



CIRCLE OR ARC GENERATOR FOR GRAPHIC DISPLAY

INTRODUCTION

Many displays operate as a Cartesian coordinate system and position a viewable trace on a display face in response to electrical signals that represent either an X-Y address to which the trace is to be moved or represent incremental changes from an existing position in the X and Y coordinate system. In the display for which a preferred embodiment of this invention has been developed, the X and Y coordinates are each divided into 1024 increments called "raster units" and each coordinate is addressable by a 10 bit binary address. Thus, a digital addressed display of this type is not easily suited for drawing circles and related geometric figures. One familiar technique for drawing circles in a display with a Cartesian coordinate system of trace positioning is to apply a sine voltage to the deflection coils or electrodes of one coordinate and a cosine voltage to the other coordinate. If the frequencies are the same and the amplitudes are the same, these inputs produce the familiar Lissajous circle pattern. However, in such systems it has been difficult to maintain equal amplitudes so as to maintain the radius equal around the circle. In addition, the use of Lissajous patterns requires an oscillator which may not be required for any other function in the display. A general object of this invention is to provide a new and improved system that produces circles or related figures in response to digital inputs of the type otherwise used in the display.

Alternatively in the prior art, problem programmers have written instructions for moving the trace in a series of short increments to form a sequence of connected chords that closely approximate a circle. Such a program might include a table of deflection values as calculated by the problem programmer himself or it might contain a set of instructions for generating these values and another set of instructions for putting the values into a form suitable for controlling the display. This procedure is undesirably complex for the problem programmer and it requires significant amount of main storage space in the data processing system for storing the tables and the related instructions.

SUMMARY OF THE INVENTION

One object of this invention is to relieve the problem programmer of the job of making up the table. The problem programmer has only to specify the radius of the circle, the center point of the circle, and several optional parameters that make the system more useful. The system of this invention is provided with a storage device that holds a table of trigonometric functions from which a series of chords forming an approximate circle can be generated as a function of the parameters supplied by the problem programmer.

Logic and timing circuits are provided for combining the information produced by the problem programmer with the information in the memory to produce electrical signals for controlling the display to producing a corresponding circle. With additional features, the circuit of this invention is useful for drawing segments of arcs, for drawing circles or segments of arcs of dashed lines, and other geometric figures.

More specifically, the problem programmer initially transmits to the display a starting X address and a starting Y address, and the display moves the trace to the

designated point in its normal mode of operation. This point is treated as the center of the circle in an operation that will be called "circle mode" and it is treated as the starting point for an arc (or a circle) in an operation that will be called "arc mode." Otherwise unused bit positions in the data bytes containing the X and Y address are used to identify that the display is to begin a circle mode or an arc mode operation. The problem programmer also arranges to transmit next a byte containing the desired radius and another byte containing the number of the chord that the system is to draw before asking for additional control information or discontinuing the circle mode or arc mode operation. In response to these inputs, the circuit of this invention accesses the memory for the trigonometric functions for the first chord, multiplies the functions by the designated radius to produce ΔX and ΔY values, and transmits the incrementing signals to the display for drawing one chord of the arc. The system continues drawing chords from point to point until it reaches the end point defined by the programmer.

One of the control bits identifies whether the circle mode or arc mode operation is to continue after the end point. If the operation is to continue, the circuit signals the channel that it is ready to accept two more bytes of the type already described for drawing a further arc.

The system is also adapted to receive an additional two bytes that specify two successive end points for two sequences of chords, one of which is blanked and the other of which is unblanked. With this feature, the system easily provides dashed line circles or arcs.

The specific embodiment of the invention is readily adaptable to various displays and to producing a variety of geometric figures.

THE DRAWINGS

FIG. 1 is a geometric figure illustrating the method for generating arcs that is used by the system of this invention.

FIG. 2 shows schematically a memory and the table of values for generating arcs.

FIG. 3 is a block diagram of a circuit for decoding data bytes transmitted to the system of this invention in response to a problem program.

FIG. 4 shows a block diagram of the preferred system of this invention.

THE EMBODIMENT OF THE DRAWING

Arc Generation FIG. 1 illustrates the operation of generating a circle as a series of 24 chords. A point 12 is the center of the circle and it is defined by conventional addressing as will be described later. The blanked beam is then moved horizontally to the right by the number of raster units that is specified for the radius by the problem programmer as shown by a line 13 and a point 14. (This direction is arbitrary but it follows a standard convention in designating angular positions in coordinates.) A second radius 16 and a point 17 show the 15° angle of the next radius. As the beam is moved from point 14 to point 17, it traces (unless it is blanked) a first chord 18. Similarly, a radius line 20 and a third point 21 illustrate another 15° increment for which the beam forms a second chord 22. The operation illustrated by FIG. 1 continues around the circle with particular chords or sequences of chords blanked or unblanked; and the operation may proceed through

24 chords for a complete circle or it may be ended at a point chosen by the problem programmer.

The calculation of the ΔX and ΔY increments for any chord of the circle is illustrated by the ΔX and ΔY increments for moving from point 17 to point 21. The 15° angle to the first radius line is 15° from the reference line 13 for this example but will be designated B for generality. Similarly, the second angle which is in fact 30° from the reference line 13 will be designated angle A. It can be seen that $\Delta X = R \cos A - R \cos B$, which can be rewritten $R (\cos A - \cos B)$. Since $\cos 15^\circ$ is larger than $\cos 30^\circ$, the term is negative as is appropriate since the trace is moved horizontally to the left in the example. The signs of the trigonometric function of course produce the proper deflection in each quadrant of the circle. Similarly, the trace is moved upward from point 17 to point 21 according to the function $R (\sin A - \sin B)$.

The advantages of dividing a circle into 15° segments are well known in various situations. For example, the system can produce commonly used fractions of a circle such as a half circle, a quarter circle, and so on. In addition, the chord subtended by a 15° angle is very close to the actual circumference of the circle and the small difference is not seriously noticeable in a display. The concepts of this invention can be easily extended to providing chords with larger or smaller angles, but smaller angles require additional memory capacity for the table and they provide no significant improvement in ordinary applications; larger angles make the figure noticeably non-circular and limit the number of useful arcs that can be drawn.

From a more general standpoint, the three points 14, 17 and 21 are arbitrary positions with respect to point 12 and consequently any desired figure of a reasonable number of segments may be drawn. Thus, the radius can be thought of as a scaling factor for such figures.

The Stored Table of FIG. 2 shows a table of the sin and cos functions explained in FIG. 1 for generating an arc of a circle. The columns $\cos A - \cos B$ and $\sin A - \sin B$ have already been explained. There is also a column for the sign of the function. Displays are commonly constructed to respond to an unsigned address and to a sign bit that controls the addition or subtraction of an incremental address. The functions and values that are stored in the memory are stored in 7 bit binary form, but for convenience, the corresponding decimal numbers are written in the table of FIG. 2. To make the terms more directly convertible to binary form, the common denominator 128 is used for each fraction; only the numerator of a fraction is shown. (The usual decimal representation of binary places to the right of a decimal point as one-half, one-fourth, one-eighth . . . can be equivalently represented as 64/128, 32/128, 16/128 . . . 1/128. Thus, a decimal term of the table such as 33 can be easily represented as 0100001 corresponding in decimal notation to $32/128 + 1/128$. Note that the numbers in the tables also equal the ΔX or ΔY increments for a radius of 128 units.)

Circuits for holding the tables of FIG. 2 are well known in various forms and are commonly called "control memories" or "read only memories." FIG. 4 shows such a memory in relation to other components of the system. Such circuits respond to a multi-bit input called an address to produce the data previously stored at the address. In the table of FIG. 2, a column has been

drawn with the addresses 1 through 24 for the 24 segment generating functions of the table. These addresses in an associatively addressed memory would represent the contents of actual data storage locations, but for more conventional storage devices, the corresponding logic is contained in an address decoder circuit (30 in FIG. 4) that selects the desired word of the memory from an address supplied on an address bus. Thus, in the sense that the storage device responds to predetermined input data (an address) and produces related output data (the addressed memory word), the table can be stored in circuits known by such names as decoders and translators.

The operation of stepping through the sequence of words in the table as corresponding arc generating operations take place in the display can be provided in various known ways. For example, a 24 bit ring can be stepped to successively address each word in sequence. The drawing illustrates the technique common with read only memories of storing the next address with the word that is being read out. The corresponding field in the data register in the memory is then used to access the next word. The actual signals for advancing to the next operation are provided by the signal that the display itself produces as a normal operation to signal that it is ready to accept a next address. As a modification to the table of the drawing, the table may include a 0 address term for generating the first increment shown in FIG. 1 as line 13 in which $\Delta X = R$, $\Delta Y = 0$.

The Data Format And The Decode Circuit of FIG. 3 The problem programmer provides instructions to the system of this invention in a format that corresponds to the requirements of a particular display. The format and the decoding circuits shown in FIG. 3 are particularly adapted to operate with a commercially available display called the 2,250 and described in a publication entitled "IBM System/360 Component Description; IBM 2250 Display Unit," Cat. No. A272701, A272702 and A272721, available from the assignee of this invention.

In such a system, a channel or other apparatus transmits data to the display as a byte of 8 bits and the byte is stored in a register or buffer memory 34. For positioning the trace at a selected point on the display face, the channel transmits four bytes (32 bits) that contain a 10 bit X address, a 10 bit Y address, and 12 bit positions that are available for various control purposes. In the normal operation, the display responds to the X and Y addresses in these bytes to move the trace to a selected point on the face of the display; when the system is in circle mode this point corresponds to point 12 in FIG. 1, the center of the circle to be drawn. For its normal operations, the display includes means for recognizing particular bit positions and byte positions in such a group of bytes.

In the preferred embodiment of this invention, a 1 in bit position 2 of the third byte signals that the circuit of this invention is to operate in circle mode. Similarly, a 1 in bit position 2 of the fourth byte of the address group identifies that the circuit of this invention is to operate in arc mode. For decoding these conditions, a latch 35 has its set input connected to an AND gate 36 to receive bit position 2 of the data register and a latch 37 is connected to an AND gate 38 to similarly receive the same bit position. An input 40, provided by the display to identify the third byte, enables gate 36 to set the latch circle in response to the circle mode bit and an

input 41 enables gate 38 to similarly respond to the arc mode bit.

The problem programmer provides two more bytes that the data processing system transmits to the display in response to a request made by the display as the normal result of completing the preceding operation and the fact that the system is in circle mode or arc mode. The bytes are identified by signals CONTROL CYTE 1 and CONTROL BYTE 2, and this group of two bytes is identified by a signal GROUP 1. Bit position 0 in these bytes identifies whether the operation is to continue beyond the arc segment. This bit would always contain a 0 if the programmer intends to draw a complete circle in one operation, and it will also contain a 0 if the programmer intends to draw only a single segment of an arc and then go to some other operation. If a 1 is held in this bit position of byte 1, a latch 40 is set and if a 1 is held in this bit position of byte 2, a latch 41 is set. Logic is provided for shifting the contents of latch 41 to latch 40 for a variation of the simple circle mode operation, as will be described later. In one useful application of the continue bit, a program can keep the trace blanked for the first arc to move the trace to a particular position on the circumference of the circle and the program can then transmit another arc defining a group of bytes to draw a visible arc from that point. Similarly, the programmer could draw a circle defined by one control word and then draw a further fraction of a circle to position the trace at any particular point on the circumference for beginning some other graphic operation.

Bits 1 through 7 of the first of these two bytes define a seven bit radius and this portion of the input data register is transferred to a separate register to be used for generating the X and Y increments. The gating is similar to the gates described for the latches CONTINUE except that the block with the legend G represents an individual AND gate for each bit of the radius word.

The second of these two bytes contains in bit position 3 through 7 a 5 bit number called END defining the last segment to be drawn and, equivalently, the last word to be fetched from the memory. Logic shown in the drawing loads the word END into a 5 bit register. Logic means similar to the logic for latches CONTINUE provides for simultaneously storing two such END words and for shifting from one to the other. Bit position 2 defines the negative (clockwise) or positive (counter clockwise) rotation for the arc. Bit position 1 defines whether the beam is blanked or unblanked. These bits are stored in latches that are similar to the latches already described. The logic and timing for these latches will be easily understood when their operations are described later. The latches of FIG. 3 can be thought of as a multi-bit register and from a more general standpoint, they form means for storing single bit parameters specified by the problem program.

As the circuit has been described so far, the following operations have taken place. The display has responded in its normal fashion to the address and has moved the trace to this position. The radius number has been loaded into register RADIUS and the number of the end point has been loaded into register END. The latches CONTINUE, ROTATION, and BEAM have been set or left reset according to the corresponding bit positions of the two control words. The system is now ready to generate the arc defined by these registers and

latches without further control from the channel. The operation just described can also be considered from the standpoint of the program that is supplied by a problem programmer. To draw a circle of a selected radius, the programmer supplies only 6 bytes to register 34 of FIG. 3. Bytes 1 through 4 contain in conventional format the address of the center of the circle, and bit 2 of byte 3 contains a 1 to signify that a circle is to be drawn. The system responds to these 4 bytes by moving the trace of the display to the specified address and by setting latch 35 in FIG. 3. Byte 5 contains the radius length in byte positions 1 through 7. Bit position 0 is insignificant and may be either a 1 or a 0. Byte 6 has a 0 in bit position 0 to signify that there are no more control bytes in this circle drawing operation. Bit position 1 contains a 0 to signify that the beam is not to be blanked. Bit position 2 contains a 1 to signify that the circle is to be drawn counter clockwise, and bit positions 3 through 7 contain the number 24 to signify that the system is to draw the 24 chords of a complete circle. The system then operates without further intervention by the problem programmer. If bit 2 of byte 3 had contained a 0 and bit 2 of byte 4 had contained a 1, the system would have operated in arc mode to draw the same circle except that the first chord would start from the point addressed in bytes 1 through 4. In the more usual arc mode operation, bit positions 3 through 7 of byte 6 would specify only a fraction of the 24 chords for the full circle of this example.

The System of FIG. 4 — Circle Mode The components and operations of the system of FIG. 4 can be understood most easily by considering that the only significant changes to the display are shown in the drawing and that in circle mode or arc mode the display communicates with the system of this invention in the same way that it would otherwise communicate with the channel or other data source. Thus, the display produces a signal REQUEST on a line 50 to signal the end of a preceding operation and to request control bytes from the channel for a next operation. In the system of FIG. 4, gates 51, 52 transmit the request directly to the channel except when the system is to receive its deflection signals from the memory.

Similarly, the channel normally transmits control bytes to a display decode circuit that produces the address and control signals that are shown in the drawing. When the system is in circle mode or arc mode, the outputs of the display decode circuit are inactive and some of the signals are produced by the system of FIG. 4. Preferably, the display decode circuit receives the signal CONTINUE which disables the decode circuit until the last byte group for a circle or arc mode operation has been received. This signal may also be used to permit operations by the display decode circuit and the channel that are independent of the circle or arc mode operation.

A gate 55 responds to the signals CIRCLE MODE and the signal GROUP 1 identifying the presence of the first control byte in the data register for transmitting the multi-bit word RADIUS to the X address input of the display. In response to this signal, the display moves the trace to the right by the distance defined by the word RADIUS and it transmits a signal REQUEST.

A gate 56 responds to the signal REQUEST, a signal NOT COMPARE that signifies that the system has not yet completed the series of chords defined by the current control bytes, and to a signal described later, to

transmit a multi-bit next address from the next address field of the memory data register to the address circuit of the memory. In response to such an address, the memory produces in its data register the function and address fields of the addressed word.

The system of FIG. 4 includes a digital multiplier 55 that receives the signal RADIUS from the circuit of FIG. 3 and the appropriate function field from the memory data register and produces outputs that are applied to the ΔX and ΔY inputs of the display. The ΔX and ΔY increments are generated separately in a sequence defined by timing signals X TIME and Y TIME. Gates 57 and 59 respond to the signal X TIME to transmit the contents of the cosine function field of the memory data register to multiplier 55 and to transmit the product produced by multiplier 55 to the display. Similarly, gates 58 and 60 transmit the contents of the sine function field of the memory data register to multiplier 55 and transmit the output of the multiplier to the display. Many suitable digital multipliers are well known. Preferably, the contents of the selected function field of the memory data register are shifted once for each bit position of the multi-bit word RADIUS and the shifted fields are accumulated where a 1 appears in the corresponding bit position of the word RADIUS.

The display responds to the signals from multiplier 55 in the same way that it would otherwise respond to signals from the channel and it again produces the signal REQUEST at the end of the operation to produce the chord defined by the multiplier output. The operation just described continues from chord to chord until the chord defined by the problem programmer in the word END is reached. A compare circuit 62 compares the current address field of the memory data register with the word END and produces NOT COMPARE, introduced as an input to gate 53, and COMPARE. When the operation has reached the end chord, the function NOT COMPARE = 0 disables gate 56 to prevent the memory from responding to the next request. The signal COMPARE operates in gates 51 and 52 to transmit the request to the channel. A channel may transmit two additional control bytes for drawing another arc or it may go to another routine. The last control byte in an arc drawing operation is identified by the CONTINUE bit in the first control byte and an AND gate 65 responds to the signals COMPARE and NOT CONTINUE to reset the next address field of the memory data register to address position 1.

The System of FIG. 4 — GROUP 2 Control Bytes The decode circuit of FIG. 3 is arranged to handle an alternate control byte format that is particularly useful for programming a dashed line circle or other figures in which the system is to continue an operation in the same mode and with the same radius but with some of the other control bits changed. A two byte group identified by a signal GROUP 2 has the format of control byte 2 of group 1 and this provides two successive end points. When the operation has reached the first point desired at the output of the decode circuit of FIG. 3, a logic network 67,68,69 routes the resulting COMPARE signal to shift the circuits of FIG. 3 and to enable gate 56 to respond to the associated REQUEST for fetching the next deflection word from the memory. When the operation reaches the point defined by the second word END, the resulting compare is routed to circuit 51 for transmitting signal REQUEST to the channel in the way already described.

The System of FIG. 4 — Arc Mode The components of the system that have been described so far can be used for drawing various noncircular configurations by appropriate arrangement of the control bytes. It is often useful to draw an arc from a predetermined point, and the system of FIG. 4 includes components that substantially simplify programming such an operation.

As has already been explained in relation to FIG. 3, the latch ARC MODE is set in response to a 1 in bit position 2 of address byte 4. The signals ARC MODE and CIRCLE MODE are combined in an OR logic function so that many of the operations described for circle mode are performed also for arc mode. Gate 55 opens only in response to the signal CIRCLE MODE and consequently the trace is not offset by the radius value as it is in circle mode.

In arc mode the system can provide functions starting at any selected point in the table so that a chord of any available angle can be formed from the previously addressed point on the display face. Logic means is provided for loading the next address field of the memory data register from the END field of control byte 2 of group 1. A group 2 byte is then transmitted with two END words as in the operation on group 2 bytes in circle mode.

Other Operations The preferred embodiment of the invention can be operated to provide a wide variety of geometric figures that are made up from chords defined in the memory and parameters defined by the problem program. The system can be readily adapted to other types of graphic devices having a Cartesian coordinate addressing system and it can be adapted to apparatus having various specific control word formats. The logic and timing functions of FIGS. 3 and 4 can be provided in many well known ways and a specific embodiment will be suggested in part by the organization of the apparatus that is to use the system of this invention. Thus, those skilled in the art will recognize various applications and appropriate modifications to the disclosed embodiment with the spirit of the invention and the scope of the claims.

What is claimed is:

1. A system for controlling a graphic device to produce one of a plurality of geometric figures composed at least in part of chords of a circle, comprising;
 - means for storing a table of functions defining a succession of chords,
 - means for decoding data bytes supplied by a problem program, including first means for decoding a byte defining an operating mode in which the system is to produce one of said figures,
 - means operable when said system is in said mode for accessing said table storing means for a function defining a next chord,
 - means connected to receive the accessed function for forming chord defining signals and means for transmitting said signals to said device, and
 - means for identifying the position of an accessed chord in said succession of chords, said decoding means including second means responsive to a set of control bits supplied by a problem program defining an end chord, and means for comparing said position of an accessed chord and said signal defining said end chord for signalling the completion of the end chord to said means for accessing said table storing means,

whereby a graphic device is controlled to form a selected figure in response to said bytes supplied by said problem program without further intervention by said problem program.

2. The system of claim 1 wherein said means for decoding data bytes comprises third means for storing a radius length word supplied by said problem program, and

said means for forming chord defining signals includes a multiplier for multiplying the radius length by a function accessed in said table to produce said chord defining signals,

whereby circles or related figures of preselected radius may be formed by said device in response to control bits supplied by said problem program defining said end, said radius, and said operating mode.

3. The system of claim 2 wherein said first means for decoding bytes produced by a problem program includes means to decode and store a first bit defining a circle mode or a second bit defining an arc mode operation called for by said problem program, and

said means to form said chord defining signals includes means operable in said circle mode before forming said first chord to control said device to move its trace in a predetermined direction by said radius length,

whereby the trace position defined by an operation immediately preceding said radius length storing operation defines the center of a circle or a partial circle to be formed.

4. The system of claim 3 wherein said decoding means includes means operable in arc mode for storing control bits defining a selected starting address in said table storing means, and

said means for accessing said table storing means includes means to begin an accessing sequence at a predetermined position in said circle mode and at said selected starting address in arc mode.

5. The system of claim 2 wherein said means for decoding includes fourth means for controlling said device according to a control bit position in a byte sup-

plied by a problem program defining a blanked or unblanked condition for said device, whereby the trace may be moved to a position defined by a sequence of chords without visible representation in the device.

6. The system of claim 5 wherein

said second means for decoding includes means operable in response to a first group of control bytes to store said end value and includes means differently responsive to a second group of control bytes to store first and second end values, and

said means for signalling the completion of an end chord includes means responsive to the completion of a chord succession defined by a first end value to begin an operation defined by the second end value and responsive to the completion of said chord defined by said second end value to produce a request to the problem program source for a next message.

7. The system of claim 6 wherein said first means for decoding bytes produced by said problem program includes means responsive to predetermined bit positions in the message format for other device operations,

and said decoding means is operable only when said system is in said mode,

whereby the trace of said device moved to a starting position for drawing one of said figures in response to a message containing a mode defining bit and thereafter responds to a subsequent message containing bits defining said end chord, said radius, and other parameters.

8. The system of claim 7 wherein said decoding means includes fifth means responsive to a bit defining whether the operation is to continue with further such control bytes after an end chord forming operation, and

said chord forming means includes means responsive to said continue bit to selectively reset or maintain the accessing position for said table storing means at the completion of an end chord forming operation.

* * * * *

45

50

55

60

65